PATENT
2042/112025-0419

UNITED STATES PATENT APPLICATION

of

Kaartik Viswanath
and
Stanley Ratliff

for

A METHODOLOGY THAT ASSURES UNIQUENESS OF DERIVED SUBORDINATE RESOURCE NAMES

A METHODOLOGY THAT ASSURES UNIQUENESS OF DERIVED SUBORDINATE RESOURCE NAMES

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to computer network systems which include mainframe devices and, more particularly, to a method for providing unique names to subordinate resources addressed by the mainframe device.

Background Information

The System Network Architecture (SNA) was developed by IBM as a means to interconnect one or more mainframe devices with computer terminals, printers, personal computers, and other devices, denoted as Physical Units (PUs). A PU may include one or more Logical Units (LUs). In the present state of the art, the LUs communicate with the mainframe device in the SNA system by means of IBM's Virtual Telecommunications Access Method (VTAM).

In the SNA system, each PU is assigned a name obtained from a resource definition file resident in the local directory of the mainframe device. The term 'resource' is used herein to refer to any network accessible unit (NAU), such as a PU, an LU, or a control point, that can be represented in the local directory. The local directory resource definition file thus contains the logical name, or resource name, and node identification for each NAU in the SNA system. In particular, the TN3270 server utilizes resource names of eight characters in length, in accordance with SNA/VTAM standards, with the requirement that each resource name be unique. By utilizing resource names in the network, an end user can begin a session as an LU without needing to know the locations of the network resources.

There has always been a problem when attempting to derive subordinate resource names from the name of the corresponding superior resource. Currently, there is no scheme by which, when attempting to do such derivation, one can assure that unique subordinate names are generated.

The TN3270 server allows the user/administrator to specify a 'user seed' to be used for subordinate resource names. If a user seed is not specified, a name seed is provided by using the first five characters of the PU name. Either seed is then used to create names for the subordinate resources by appending to the seed an index value ranging from 1 to 255.

Such conventional derivation methods for name seeds cause clashes of the subordinate resource names in the name space. By way of example, a clash will most likely result when the first five characters in the PU names remain invariant.

What is needed is a method for deriving subordinate resource names which insures that the derived names are unique.

SUMMARY OF THE INVENTION

The aforementioned shortcomings of the present art are addressed by a method for generating a unique subordinate resource name. The disclosed method comprises the steps of identifying a subordinate resource and a related superior resource; ascertaining the name of the superior resource; truncating the superior resource name to form a truncated name; obtaining a counter number from a global counter; appending the counter number to the truncated name to form an appended name; and assigning the appended name to the subordinate resource. In an alternative embodiment, the method includes the steps of identifying the subordinate resource and the related superior resource; ascertaining the name of the superior resource; obtaining a counter number of n digits from the global counter; substituting the counter number for n characters in the superior resource name to form a derived name; and assigning the derived name to the subordinate resource.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

Fig. 1 is a functional representation of a computer system network in accordance with the present invention;

Fig. 2 is a diagrammatical representation of a portion of the computer system network of Fig. 1;

Fig. 3 is a flow diagram illustrating a preferred embodiment of the inventive process; and

Fig. 4 is a flow diagram illustrating an alternative embodiment to the method of Fig. 3.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

There is shown in Fig. 1 a computer system network 10 including an application 20 utilized by one or more resources. In the illustration, the application 20 is accessed by a first resource 40 and a second resource 50 through a k^{th} resource 70. There may be one or more resources subordinate to each of the resources 40, 50, and 70. In the example provided, the first resource 40 includes a first subordinate resource 41 (a_1) and a second subordinate resource 43 (a_2) through an m^{th} subordinate resource 45 (a_m). Similarly, the second resource 50 includes a first subordinate resource 51 (b_1) and a second subordinate resource 53 (b_2) through a p^{th} subordinate resource 55 (b_p). A global counter 30 communicates with all of the resources 40 and 50 through 70.

In a preferred embodiment, a computer system network 100 includes a mainframe device 110 and a server 120, as shown in Fig. 2. Preferably, the computer system network 100 conforms to SNA/VTAM standards. The mainframe device 110 communicates with the server 120, which is preferably a TN3270 server. The server 120 communicates with a plurality of resources, here denoted as a first physical unit 140 (PU1), a second physical unit 150 (PU2), and a third physical unit 160 (PU3) through a j^{th} physical unit 190 (PUJ). Each physical unit 140 through 190 includes one or more logical units. In the example provided, the first physical unit 140 includes a logical unit 141 (LU1) and a logical unit 143 (LU2) through a logical unit 149 (LUN1). The second physical unit 150 includes a logical unit 151 (LU1). The jth physical unit 190 includes a logical unit 191 (LU1) through a logical unit 199 (LUNJ).

In a conventional resource-naming system, the first physical unit 140 may be assigned a name such as 'HERESPU1' and the second physical unit 150 may have the name 'HERESPU2.' In a conventional subordinate name derivation procedure, the subordinate name seed is derived from the first five characters, 'HERES,' of the name of the first physical unit 140. The resulting logical unit names derived from the name seed 'HERES' will be 'HERES001' and 'HERES002' through 'HERES255.' The logical unit 141, for example, will then have the name 'HERES001.'

However, since the subordinate name seed derived from the first five characters of the name of the second physical unit 150 will also be 'HERES,' the logical unit 151 will also have the name 'HERES001.' That is, two different subordinate resources, 141 and 151, are both assigned the same resource name. This violates the SNA/VTAM requirement that each resource name be unique.

The disclosed method is best explained with reference to Fig. 3 in which the global counter 130 is utilized to provide a three-digit prefix for inclusion with a retrieved resource name. In a preferred embodiment, the relevant resource name is obtained, in box 201. The last three characters of the resource name is truncated, in box 203, to provide the name seed. For example, truncation of the resource name of the first physical unit 140, 'HERESPU1' will yield 'HERES.' The current counter number is obtained from the global counter 130, in box 205.

Initially, the counter 130 is set to '000.' This number is appended to the truncated resource name, in box 207. The appended name is assigned to the subordinate resource, in box 209. For the above example, the derived name provided for the logical unit 141 will be '000HERES.' If additional subordinate resource names are to be derived, at box 211, the global counter 130 is incremented, at box 213, and the process returns to obtain the new, current global counter value, in box 205. In the example, the global counter 130 is incremented to ‘001’ and the derived name for the second logical unit 143 will be ‘001HERES.’

In an alternative method of deriving resource names, a global counter providing four-digit counter numbers is used. Accordingly, the last four characters of the resource name are dropped, and the four-digit counter number is appended to the truncated resource name. Using the above example in which the first physical unit 140 has been assigned ‘HERESPU1’ and the second physical unit 150 has been assigned ‘HERESPU2,’ the last four characters are truncated to give ‘HERE,’ and the corresponding subordinate names for the first physical unit 140 and the second physical unit 150 will be ‘0000HERE’ and ‘0001HERE,’ respectively.

It should be understood that the disclosed method is not limited to use with a counter providing three- or four-digit counter numbers, but may be advantageously used with counters providing five-digit counter numbers. Using the physical unit names of the above examples, derived resource names of “00000HER’ and ‘00001HER’ may be obtained, for example.

In yet other alternative embodiments, counter numbers provided by a three-digit counter may be substituted for three sequential characters anywhere in the character strings of the physical unit names. For example, the second, third, and fourth characters can be replaced by the global counter to give the subordinate name ‘H000SPU1’ to the first physical unit 140 and the subordinate name ‘H001SPU2’ to the second physical unit 150. Or, the, third, fourth, and fifth characters can be replaced to give the subordinate name ‘HE000PU1’ to the first physical unit 140 and the subordinate name ‘HE001PU2’ to the second physical unit 150. Thus, all applications using these resource names as references to generate names will differ in at least three of the first five characters. In general, any three of the eight characters in a physical unit name can be replaced by the three-digit counter number. However, in a computer system network including a TN3270 server, the last three characters in a physical unit name are not replaced.

These alternative methods are illustrated in Fig. 4 where in which the superior resource name is obtained, in box 215, and the global counter 130 is utilized to provide an n-digit counter number, in box 217. The current counter number is substituted in the physical unit name, in box 219. The resulting name is assigned to the subordinate resource, in box 221. If additional subordinate resource names are to be derived, at box 223, the global counter 130 is incremented, at box 225, and the process returns to obtain the new, current global counter value, in box 217.

What is claimed is: